มุม dBASE

ดีเบส กับ เอ็ม เอสดอส

ยืน ภู่วรวรรณ

ผมได้รับคำถามจากผู้สนใจคอลัมน์นี้ถึงวิธีการสร้างเมนูในระดับเอ็มเอสดอส ลองดูคำถามต่อไปนี้

"เครื่องคอมพิวเตอร์ที่ผมใช้เป็นเครื่องที่เหมือนกับไอบีเอ็ม มีฮาร์ดดิสค์ขนาด 10 MB ปรากฏว่ามีผู้มาใช้เครื่องร่วมกันหลายคน ส่วนใหญ่ใช้โปรแกรมสำเร็จรูป เช่น เวิร์ดสตาร์ ดีเบส โลตัส ขวัญใจนักพิมพ์ดีด ผมได้ทำการสร้างไดเรคเตอรีย่อย ตอนสร้างฟอร์แมทดิสค์ใหม่ๆ โดยแยกเป็นไดเรคเตอรีของเวิร์ดสตาร์, ดีเบส ฯลฯ โดยใช้

C > MD WS	สร้างไดเรคเตอรีเวิร์ดสตาร์
C > CD WS	เรียกเข้าไดเรคเตอรีเวิร์ดสตาร์
C > COPY A:W*.* C:	ก๊อบปี้เวิร์ดสตาร์จากไดรฟ์ A มาไว้ในไดเรคเตอรีนี้
C > CD	ออกไปเมนไดเรคเตอรี

ตราวนี้ผมมีไดเรคเตอรีย่อยบนฮาร์ดดิสค์ ผู้ใช้จะใช้จึงดูยุ่งยาก ผมอยากให้ผู้ใช้เปิดเครื่องจะปรากฏเมนูดังรูปที่ 1

```
MENU
1 WS                    2 DBASE
3 THAIEASY              4 DOS
5 LOTUS                 6 SC3
ENTER YOUR SELECTION:
```

รูปที่ 1 เมนเมนูหลังจากฟอร์แมทใหม่แล้ว

ถ้าผู้ใช้เลือก 1 เพียงกดคีย์เลข 1 โปรแกรมจะวิ่งเข้าเวิร์คสตาร์เอง แต่เมื่อเลิกใช้เวิร์คสตาร์แล้วจะกลับมาเมนูนี้เอง การเลือกเมนูเมื่อรันโปรแกรมนั้นแล้ว และ QUIT ออกมาจะปรากฏเมนูนี้เสมอ

เป็นไงครับปัญหาของแฟนมุมดีเบสนี้ ที่อยากแสดงเมนูในระดับ DOS

แนวความคิด

ในตอนแรกผมคิดว่าจะเขียนโปรแกรมด้วยภาษาแอสเซมบลี้ แล้วใส่เป็นยูทิลิตี้ให้รันในโหมด AUTOEXEC.BAT โดยอาจจะเขียนเป็น START.COM ในนี้จะเป็นโปรแกรมจัดการการทำงานทั้งหมด แต่ผมเห็นว่าการเขียนโปรแกรมด้วยภาษาแอสเซมบลี้จะยุ่งยากและผู้ต้องการจะแก้ไขเมนูหรือแก้ไขโปรแกรมจะทำเองได้ยาก ประจวบกับผมได้รับการเร่งให้เขียนต้นฉบับมุมดีเบสผมจึงตัดสินใจที่จะนำเอาดีเบสมาเขียนเมนูในระดับ DOS ท่านอย่าเพิ่งแปลกใจนะครับว่าจะทำได้อย่างไร การทำได้นี้อาศัยขีดความสามารถของเอ็มเอสดอส ผู้สนใจอาจหาอ่านรายละเอียดของเอ็มเอสดอสได้จากหนังสือที่ผู้เขียนได้เขียนร่วมกับอาจารย์สุรศักดิ์ สงวนพงษ์

ดีเบสร่วมกับเอ็มเอสดอส

บนเอ็มเอสดอสมีแบทช์ไฟล์ที่จะกำหนดลำดับการทำงานในระดับเอ็มเอสดอสได้ โดยเมื่อเปิดเครื่องจะมีลำดับการทำงานดังรูปที่ 2

จะเห็นว่าเรามีทางที่จะสร้าง AUTOEXEC เพื่อให้เกิดการสร้างเมนูให้ผู้เลือกใช้ได้ ในระดับนี้สามารถใช้คำสั่ง DOS เขียนเป็นแบทช์ไฟล์เสมือนเป็นโปรแกรม แต่คำสั่งสร้างสกรีนเป็นเมนู และรับคีย์บนเอ็มเอสดอสไม่มีโดยตรง เมื่อเป็นเช่นนี้เราจะเข้าไปทำในดีเบส เพราะเขียนโปรแกรมได้ง่าย

ในคำสั่งของแบทช์มีคำสั่งหนึ่งที่น่าสนใจ คือ IF โดยมีรูปโครงเป็นดังนี้

IF EXIST (ชื่อไฟล์) GOTO (เลเบล)

เราจึงให้ดีเบสสร้างสถานะให้ตรวจสอบได้ ถ้าตรวจสอบพบเงื่อนไขก็จะกระโดดไปทำตามเลเบลที่กำหนดได้

ลำดับของแบทช์ไฟล์ จะเป็นดังรูปที่ 3

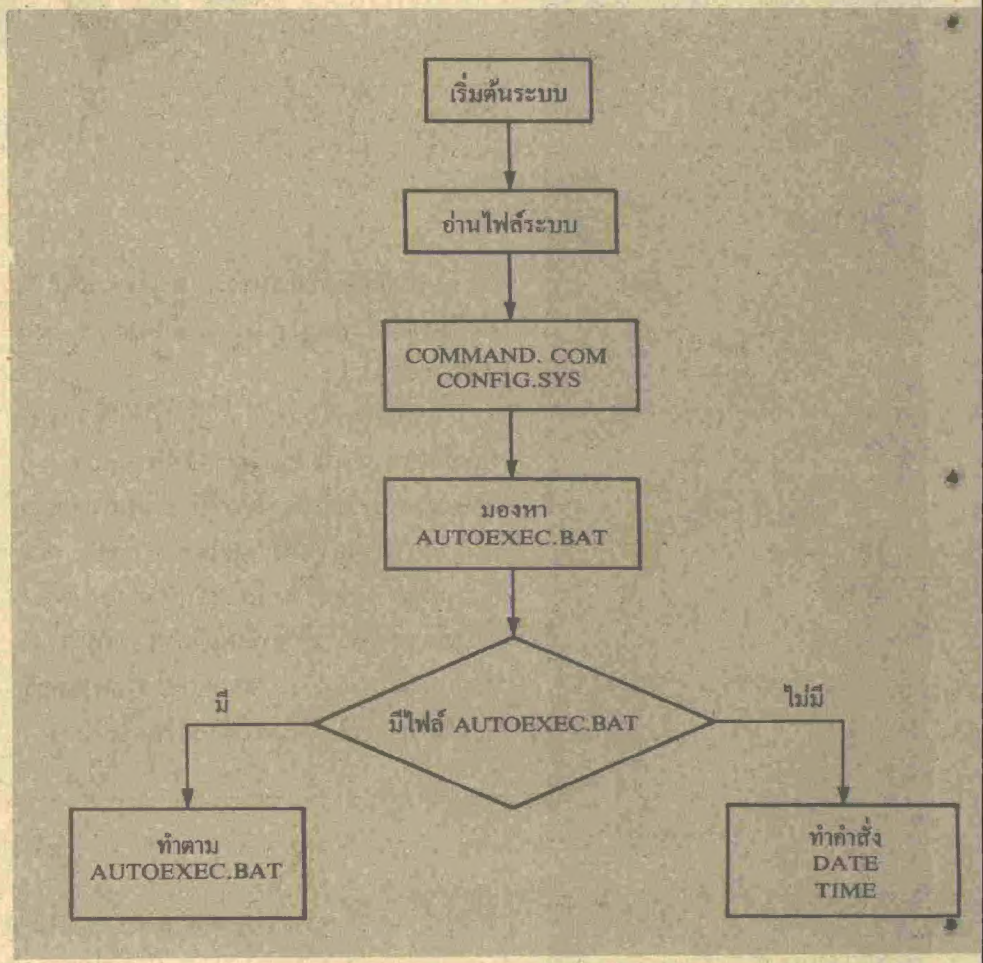
จากผังงานที่เขียนในรูปแบบแบทช์ไฟล์นี้ เราจะลูปกลับโดยเรียก AUTOEXEC ใหม่ แต่อย่างไรก็ดี เนื่องจากการทำงานในแต่ละแพ็คเกจอยู่ต่างไดเรคเตอรีกัน ดังนั้นเราจะใช้ CD คือออกมาเมนไดเรคเตอรี แล้วเรียกกลับเข้าไปโดยใช้ CD THAIEASY คือเรียกเข้าไปยังไดเรคเตอรี THAIEASY แล้วรัน THAIEASY แต่เนื่องจาก THAIEASY เป็นแบทช์ไฟล์

โปรแกรมแบทช์ไฟล์ที่ชื่อว่า AUTOEXEC.BAT ได้รับการเขียนด้วย MODIFY COMMAND AUTOEXEC.BAT หรือจะใช้เวิร์คสตาร์เขียนก็ได้ โปรแกรม AUTOEXEC.BAT เขียนได้ดังรูปที่ 4

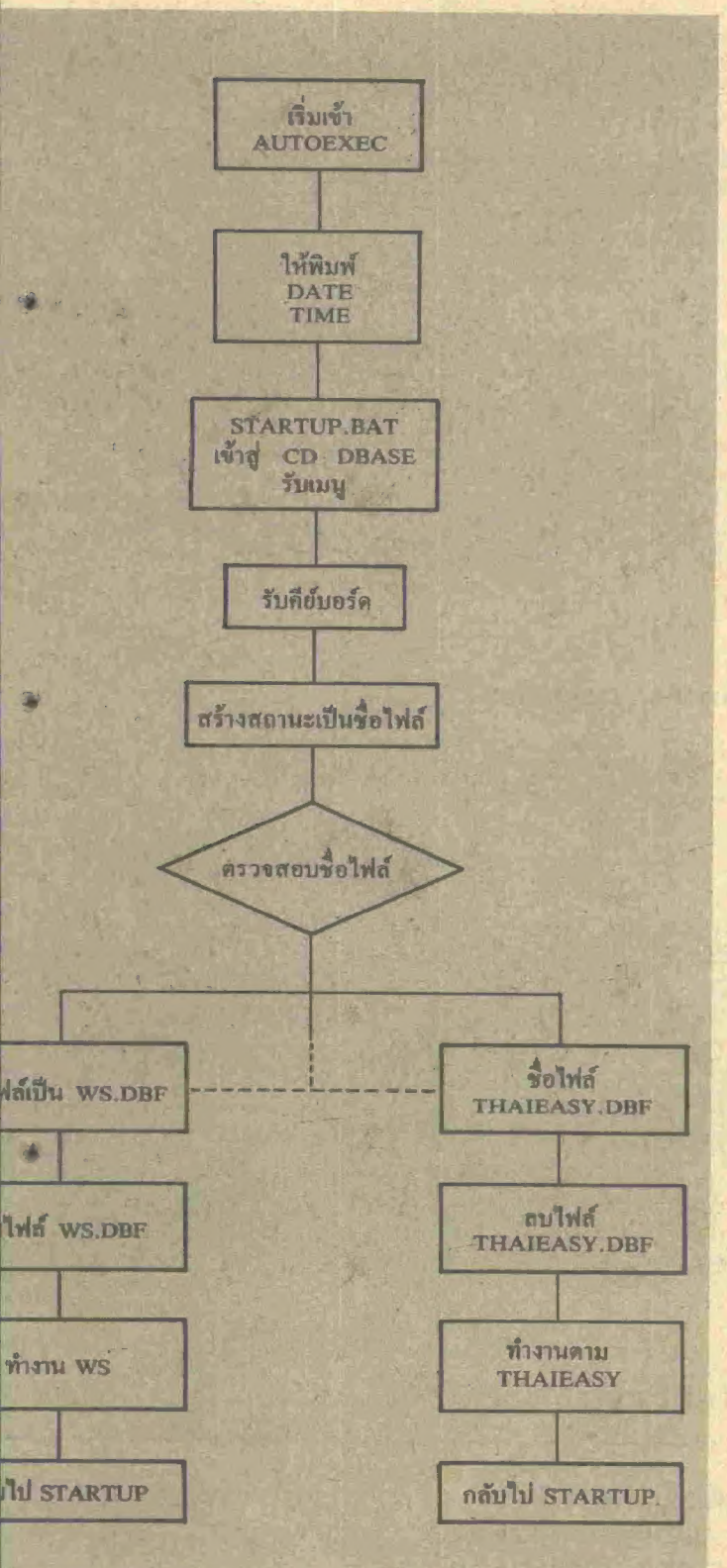
จะเห็นที่โปรแกรม AUTOEXEC.BAT นั้นเราให้ใส่วันที่และเวลา เมื่อใส่เสร็จแล้วเราจะเรียกแบทช์ไฟล์อีกโปรแกรมหนึ่งที่ชื่อว่า STARTUP

ลองพิจารณาโปรแกรม STARTUP.BAT ในรูปที่ 5

โปรแกรม STARTUP นี้ จะวิ่งเข้าไดเรคเตอรีของดีเบสด้วย CD DBASE แล้วรัน dBASE START คือทำงานตามโปรแกร



รูปที่ 2 ผังงานแสดงลำดับการทำงานของแบทช์ไฟล์ในเอ็มเอสดอส



ที่ 3 ผังงานแสดงลำดับการทำงานของแบทช์ไฟล์ในคำสั่ง IF

```
A>
A>TYPE AUTOEXEC.BAT
REM:This program is autoexec.bat
date
time
startup
```

รูปที่ 4 โปรแกรมแบทช์ไฟล์ AUTOEXEC. BAT

```
A>TYPE STARTUP.BAT
REM: This file is startup.bat
echo off
cls:
cd dbase
dbase start
if exist ws.dbf goto lab1
if exist dos.dbf goto lab2
if exist dbase.dbf goto lab3
if exist thaieasy.dbf goto lab4
startup
:lab1
erase ws.dbf
cd \
cd ws
ws
cd \
startup
:lab3
erase dbase.dbf
cd \
cd dbase
dbase
cd \
startup
:lab4
erase thaieasy.dbf
cd \
cd thaieasy
thaieasy
cd \
startup
:lab2
erase dos.dbf
cd \
cd dos
echo on
REM: eof
```

รูปที่ 5 โปรแกรม STARTUP.BAT ที่ใส่วันที่ และเวลา

START.PRG เพื่อแสดงเมนู โปรแกรม START ในดีเบสจะสร้างไฟล์ .DBF เพื่อนำมาตรวจสอบใน IF โดยใช้ IF EXIST แล้วกระโดดไปยังเลเบลต่างๆ เพื่อทำงาน

พิจารณาไดเรคเตอรีหลักจะเห็นสับไดเรคเตอรีที่สร้างไว้มี DOS, WS, THAIEASY และ DBASE ดังรูปที่ 6

เรียกเข้าไดเรคเตอรีย่อย THAIEASY โดยใช้ CD หรือการเข้าสู่สับไดเรคเตอรี ดังรูปที่ 7

จะเห็นว่าในแบทช์ไฟล์ STARTUP มีเรียก THAIEASY ถ้าผู้ใช้เลือกข้อ 4 ตามเมนู คราวใดก็ตามที่มีการเรียกแบทช์ไฟล์ตัวใหม่ มันจะเคลียร์ของเก่าทิ้ง ดังนั้นจึงต้องสร้างแบทช์ไฟล์ให้กลับได้ ในที่นี้สร้างแบทช์ไฟล์สำหรับ THAIEASY โดยการใช้ CD/ และ STARTUP ดังนี้

```
A>TYPE THAIEASY.BAT
kukey
thaimain
cd \
startup
```

คราวนี้มาดูโปรแกรม START.PRG ที่เป็นโปรแกรมสร้างเมนูในดีเบส ซึ่งเขียนด้วยดีเบส สำหรับผู้มีจอสีจะใช้ SET COLOR TO สี, สี ก็ได้ จะได้เมนูเป็นแบบเทคนิคคัลเลอร์สวยงามดี โปรแกรมที่เขียนด้วยดีเบส จะมีเทคนิคในส่วนของการก็อบปี้ไฟล์ DUMMY.DBF ในชื่อตามการเลือกเพื่อจะนำไปใช้ในการตรวจสอบในระดับเอ็มเอสดอส เมื่อตรวจสอบเสร็จก็จะลบออก ดังนั้นในไดเรคเตอรีของดีเบสนี้ จะต้องสร้างไฟล์ดัมมี่นี้ไว้ ในที่นี้สร้างโดยไม่ต้องใส่ข้อมูลและอาจมีเพียงฟิลด์เดียวก็ได้ โปรแกรม START.PRG เป็นดังรูปที่ 8

โครงสร้างไฟล์ของดัมมี่ DUMMY.DBF เป็นดังรูปที่ 9

รูปที่ 6 แสดงสับไดเรคเตอรี

```
A>DIR

 Volume in drive A has no label
 Directory of  A:\

DOS             <DIR>          1-01-80  12:02a
WS              <DIR>          1-01-80  12:03a
THAIEASY        <DIR>          1-01-80  12:05a
DBASE           <DIR>          1-01-80  12:10a
COMMAND  COM     17664          3-08-83  12:00p
AUTOEXEC BAT       128         12-12-85   8:04p
STARTUP  BAT       512         12-12-85   8:06p
ANSI     SYS      1664          3-08-83  12:00p
CONFIG   SYS        19         12-12-85   8:08p
        9 File(s)          6144 bytes free
```

รูปที่ 7 แสดงสับไดเรคเตอรีใน THAIEASY

```
A>CD THAIEASY

A>DIR

 Volume in drive A has no label
 Directory of  A:\THAIEASY

.               <DIR>          1-01-80  12:05a
..              <DIR>          1-01-80  12:05a
THAIMAIN EXE     29117         12-04-85   8:21p
THAIEASY BAT        32          1-01-80  12:15a
THAIOVR  EXE     37533         12-04-85   8:06p
BASRUN   EXE     21504          1-01-80   1:22a
KUKEY    COM      9402          1-01-80  12:03a
        7 File(s)          6144 bytes free
```

รูปที่ 8 โปรแกรม START.PRG

```
A>TYPE START.PRG
*start for autoexec.bat
*When you turn on the power
*Microcomputer will start to execute
 this program first.
```

```
*This program will create the file for
  status check
*
SET talk off
SET bell off
store 0 to answer
DO while answer <1 .or. answer >4
   erase
   set color to 2,5
   @ 5,35 say "MENU"
   @ 6,35 say "----"
   set color to 4,6
   @ 10,20 say "1. WORDSTAR"
   @ 10,50 say "2. DOS"
   @ 14,20 say "3. DBASE"
   @ 14,50 say "4. THAIEASY"
   store 0 to answer
   @ 18,20 say "Enter your selection..."
   get answer picture "9"
   read
   use dummy
   do case
      CASE answer = 1
         copy stru to ws
         quit
      CASE answer = 2
         copy stru to dos
         quit
      CASE answer = 3
         copy stru to dbase
         quit
      CASE  answer = 4
         copy stru to thaieasy
         quit
    endcase
enddo
*end of menu.prg
```

```
                    MENU
                    ----

     1. WORDSTAR                    2. DOS


     3. DBASE                       4. THAIEASY


     Enter your selection...:0:
```

รูปที่ 10 แสดงผลบนจอภาพเมื่อรันโปรแกรม START.PRG

```
.
. LIST STRUCTURE
STRUCTURE FOR FILE:   B:DUMMY    .DBF
NUMBER OF RECORDS:    00000
DATE OF LAST UPDATE:  01/01/80
PRIMARY USE DATABASE
FLD        NAME        TYPE WIDTH    DEC
001        A              C    005
** TOTAL **                    00006
```

รูปที่ 9 โครงสร้างไฟล์ DUMMY.DBF

เมื่อรันโปรแกรม START นี้ ผลจะได้ภาพบนจอดังรูปที่ 10

สำหรับตัวโปรแกรม START.PRG นั้นเกือบไม่ต้องอธิบายก็ได้ เพราะค่อนข้างจะพบเห็นรูปโครงกันดีอยู่แล้ว เพราะเป็นโครงสร้างการสร้างเมนู และรันคีย์บอร์ด

เมื่อสร้างเสร็จแล้วให้ไฟล์ AUTOEXEC.BAT อยู่ในเมนไดเรคเตอรีและไฟล์ START.PRG และ DUMMY.DBF อยู่ในไดเรคเตอรีดีเบส เสร็จแล้วลองรีเซตเครื่องดู ป้อนวันที่ เวลา แล้วจะเห็นเมนู

ดีเบสกับดีไวซ์ไดรฟ์เวอร์ในเอ็มเอสดอส

บนเอ็มเอสดอสมีวิธีการติดตั้งดีไวซ์ไดรฟ์เวอร์ได้ ทำให้ผู้ใช้สามารถติดตั้งอุปกรณ์เฉพาะบางอย่างเพื่องานพิเศษบางอย่างได้ รายละเอียดของดีไวซ์ไดรฟ์เวอร์สามารถหาอ่านได้จากคู่มือเอ็มเอสดอส 2.0 หรือจากหนังสือเอ็มเอสดอสในบทที่ 9 ซึ่งกล่าวถึง ANSI.SYS ซึ่งเป็นมาตรฐานการกำหนดเป็นมาตรฐานของอเมริกัน ซึ่งเราจะต้องมีไฟล์ที่ชื่อว่า CONFIG.SYS ซึ่งอยู่ในดอส และเราสร้างไฟล์นี้เองได้ โดยไฟล์ CONFIG.SYS จะมีการเขียนเป็นข้อความซึ่งอาจจะ

สร้างโดยใช้

C > COPY CON: CONFIG.SYS
DEVICE = ANSI.SYS
^Z กด CTRL พร้อมกับ Z
ตามด้วยรีเทิร์น

ไฟล์ ANSI.SYS จะต้องมีอยู่ในดิสก์นี้ด้วย เมื่อเริ่มบูทเครื่องจะทำการติดตั้ง ANSI นี้เข้าไว้เพื่อให้ใช้คำสั่งควบคุมจอภาพนี้ได้

โปรแกรมที่เขียนด้วยดีเบสนี้นำมาจากพายโอไมค์ แคมเบลและเซมซาเวดจ์ ซึ่งเขียนไว้ในจดหมายข่าวของบริษัทแอชตันเทต ผู้เขียนจึงถือโอกาสนำมาให้เห็นเป็นแนวทางสำหรับผู้ที่อยากจะติดต่อกับดอส ในส่วนของตัว ANSI (ขวัญใจนักพิมพ์ดีดก็เขียนโดยใช้มาตรฐานคำสั่งการจัดจอภาพด้วย ANSI เพื่อให้เป็นมาตรฐานเมื่อไปใช้กับเอ็มเอสดอสต่างฮาร์ดแวร์กัน)

โปรแกรมติดต่อกัน ANSI นี้ แบ่งออกเป็น 3 ส่วน คือส่วนแรกเป็นส่วนที่ใช้คำสั่ง POKE เพื่อนำเอาภาษาเครื่องลงในหน่วยความจำตอนบนของดีเบส

ส่วนที่ 2 คือ CHR(27) + "[O;82;22p" นี้คือ การเปลี่ยนคีย์บอร์ดจากคีย์ตำแหน่ง INS (ด้านฟังก์ชันคีย์ทางขวา) ให้มีค่าเป็นคีย์ ^V หรือกด CTRL-V พร้อมกัน เพื่อให้ใช้ฟังก์ชันคีย์นี้ได้ ดังนั้นเราสามารถสร้างคีย์บอร์ดโดยใช้ ANSI นี้ได้ (ดูจากหนังสือเอ็มเอสดอสหน้า 118 จากสำนักพิมพ์ซีเอ็ด)

เมื่อได้ทำการ STORE ค่าลงตัวแปร เช่น STORE CHR(27) + "[O;82;22p" TO INS แล้วเราใช้ CALL INS จะทำให้โปรแกรมย่อยได้รับการทำงานติดตั้งคีย์ INS นี้ และดีเบสจะรับรู้ได้ตลอดไป ถ้ายังไม่ปิดเครื่อง

ส่วนท่อนที่ 3 เป็นการกำหนดสี โดยใช้รหัสคำสั่งใน ANSI โดยถ้าผู้ใช้สั่ง CALL ตามด้วยตัวแปร ก็จะทำให้เกิดสีตามต้องการ โดยสีจะแยกเป็น สีฟอร์กราวนด์ และสีแบล็คกราวด์ ดังรูปที่ 11

```
A>
A>TYPE CONFIG.PRG
*pc MSDOS config program
*
*Requires booting with a CONFIG.SYS file containing
*the command: 'DEVICE = ANSI.SYS'
*both files ANSI.SYS and CONFIG.SYS must be present
*on the boot disk.
*
SET TALK OFF
*
*poke the assembly language subroutine into high memory
POKE 56800,80,83,82,50,237,138,15,67
poke 56808,138,23,180,02,205,33,67,226
POKE 56816,247,90,91,88,195
SET CALL TO 56800
*
*place the key redefining string in a memvariable
*and call that memvariable
STORE CHR(27) + "[O;82;22p" TO ins
STORE CHR(27) + "[O;83;127p" TO del
CALL ins
CALL del
*place the code that the DOS clear screen in a
 memvariable..
STORE CHR(27) + "[2J" TO cls
*
*set up the screen parameters into mem variable..
STORE CHR(27) + "[0m" TO normal
STORE CHR(27) + "[1m" TO bold
STORE CHR(27) + "[4m" TO underscore
STORE CHR(27) + "[5m" TO blinking
STORE CHR(27) + "[7m" TO reverse
*
*forground color
STORE CHR(27) + "[30m" TO f:black
STORE CHR(27) + "[31m" TO f:red
STORE CHR(27) + "[32m" TO f:green
STORE CHR(27) + "[33m" TO f:yellow
STORE CHR(27) + "[34m" TO f:blue
STORE CHR(27) + "[35m" TO f:margenta
STORE CHR(27) + "[36m" TO f:cyan
STORE CHR(27) + "[37m" TO f:white
*
*set blackground color
STORE CHR(27) + "[40m" TO b:black
STORE CHR(27) + "[41m" TO b:red
STORE CHR(27) + "[42m" TO b:green
STORE CHR(27) + "[43m" TO b:yellow
STORE CHR(27) + "[44m" TO b:blue
STORE CHR(27) + "[45m" TO b:margenta
STORE CHR(27) + "[46m" TO b:cyan
STORE CHR(27) + "[47m" TO b:white
*
*restore the environment if you changed it in this file.
SET TALK ON
*
*close this command by returning to the
*command that call it
RETURN
```

รูปที่ 11 โปรแกรม CONFIG.PRG

ส่วนของโปรแกรมภาษาเครื่องแสดง
ดังรูปที่ 12

เมื่อจะใช้สีลองใช้คำสั่ง

```
CALL f:margent
CALL b:blue
CALL CLS
```

เถมท้ายด้วยโปรแกรม binary search

ความจริงโปรแกรมนี้ ไม่เกี่ยวกับดอส แต่อาศัยหลักการของการค้นหาข้อมูลแบบ binary search มาค้นหาข้อมูลในไฟล์ .DBF ให้เร็วขึ้นเพราะไฟล์ .DBF เราค้นหาด้วย LOCATE ซึ่งจะค้นหาแบบลำดับและไฟล์ DBF นั้น ในบางครั้งเราได้ทำการจัดเรียงข้างฟิลด์ไว้แล้ว ตัวอย่างประยุกต์นี้เป็นการค้นหาข้อมูลในฟิลด์ที่เราได้จัดเรียงไว้แล้ว การค้นหาแบบไบนารีนี้เป็นอัลกอริทึมการค้นหาแบบง่ายๆ โดยใช้หลักการเทียบข้อมูลบ่งครึ่งไปเรื่อยๆ ลองดูโปรแกรมในรูปที่ 13

การค้นหาข้อมูลนี้ทำได้โดยการเรียกโปรแกรม BIN ด้วยคำสั่ง

DO BIN

โปรแกรมจะถามชื่อไฟล์ ชื่อฟิลด์ และค่าที่จะค้น จากนั้นจะค้นหาให้ ถ้าพบตัวแปร is:found จะเป็นลอจิกจริง และค่าเรคอร์ดพอยท์เตอร์ (# ในดีเบสทู) และ RECND

```
A>
A>TYPE CONFIG.ASM
CODE        SEGMENT PARA PUBLIC 'CODE'
            ASSUME CS:CODE,DS:CODE,ES:NOTHING,SS:NOTHING

START:
            PUSH    AX          ;SAVE REGISTER
            PUSH    BX          ;
            PUSH    DX          ;
            XOR     CH,CH       ;CLEAR HIGH 1/2 OF CX
            MOVE    CL,[BX]     ;STRING LENGTH INTO CL (CX)
            INC     BX          ;POINT TO FIRST CHARACTER
                                ; OF STRING
LOOPA:
            MOV     DL,[BX]     ;
            MOV     AH,002H     ;SUB FUNCTION DISPLAY OUTPUT
            INT     021H        ;CALL STANDARD FUNCTION
            INC     BX          ;INCREMENT POINTER
            LOOP    LOOPA       ;REPEAT FOR LENGTH OF STRING

DONE:
            POP     AX          ;RESTORE REGISTER
            POP     BX          ;
            POP     DX          ;
            RET                 ;RETURN

CODE        ENDS
            END     START       ;WHERE TO START
```

รูปที่ 12 โปรแกรมภาษาเครื่องของ CONFIG. PRG

( ) ในดีเบสทรีจะหยุดที่ตำแหน่งที่พบ แต่ถ้าไม่พบค่าของเรคอร์ดพอยท์เตอร์จะหยุดไม่แน่นอน

ส่งท้าย

ในมุมดีเบสเล่มที่แล้วได้กล่าวว่าจะเขียนเรื่องการใช้ดีเบสกับแอสเซมบลี้ แต่ต้องขอเลื่อนไปโอกาสหน้าอดใจรอก่อนครับ เพราะความจริงแล้วเราก็ใช้เชื่อมกับแอส-เซมบลี้มาหลายตอนแล้ว ฉบับนี้ก็มีการเชื่อมกับดีไวซ์ ANSI แล้วพบกันใหม่ในมุมดีเบสฉบับหน้า

ปที่ 13 โปรแกรม binary search

```
A>TYPE BIN.PRG
*BINARY SEARCH
*when the key is in sequential order
*
*is:found = .T. if a match occurs, current record is match.
*is:found = .F. if no match, current record is varies
*
erase
set talk off
store f to is:found
*
```

```
*Prompt operator for detail of the search...
ACCEPT "Enter filename  -->" to file:name
ACCEPT "Enter fieldname  --->" to field:name
INPUT "Enter data to find --->" to search:key
*Note that the input command required character type
*data to be delimited with quote marks when entered.
*
*Open the file...
USE &file:name
*
*Branch for first or last record...
IF &field:name = search:key
   *
   *Found it the first record.
   Store T to is:found
ELSE
   *May be it is the last record...
   go bottom
   If &field:name = search:key
      *
      *found it in the last record...
      store t to is:found
   ELSE
      *not there either, so let search
      *set the top,middle and bottom marker
      store # to high
      store 0 to mid
      store 1 to low
      *
      *Establish the loop for repetition..
      DO WHILE .not. is:found
         IF mid = low+INT((high-low)/2)
            SET talk on
            RELEASE file:name,search:key,field:name,low,middle,high
            RETURN
         ELSE
            STORE low+INT((high-low)/2) to mid
            go mid
            DO CASE
               CASE &field:name>search:key
                  STORE mid to high
               CASE field:name < search:key
                  STORE mid to low
               OTHERWISE
                  STORE t to is:found
            ENDCASE
         ENDIF
      ENDDO
   ENDIF
endif
RELEASE file:name,search:key,field:name,low,mid,high
RETURN
```